

AN15135

Power Management for MCX E31

Rev. 1.0 — 21 September 2026

Application note

Document information

Information	Content
Keywords	AN15135, MCX E31x, MCX E31B, Run mode, Standby mode, low power, FRDM-MCXE31B, MCX E315, MCX E316, MCX E317, wake-up
Abstract	This application note describes the Run and Standby power model of the MCX E31 family and explains the Standby entry and recovery flow implemented by the FRDM-MCXE31B <code>power_mode_switch_mcxe</code> low-level example in MCUXpresso SDK 26.06.00. It provides wake-up, state-retention, I/O, and troubleshooting guidance, and reports representative JP1 and JP2 Standby current measurements obtained with the unmodified example on FRDM-MCXE31B.



1 Introduction

For application software, the MCX E31 device supports two Power modes: Run and Standby modes. The MCUXpresso SDK example uses these modes as its two selectable power modes. Despite this compact model, a low-power transition still coordinates the processor, clocks, memory, peripherals, I/O, and wake-up sources. It is not equivalent to executing an isolated `WFI` instruction.

This application note explains:

- 1. The Run and Standby power model and the resources that remain available in Standby.
- 2. The entry and recovery sequence implemented by the MCUXpresso SDK low-level example.
- 3. How to select a wake-up source, retain the required state, and avoid common low-power integration errors.
- 4. How to obtain representative board-level Standby current measurements on FRDM-MCXE31B without modifying the SDK example.

The measurements are intentionally board-oriented. FRDM-MCXE31B is not converted into a data sheet characterization fixture. Data sheet values are used as contextual references only after the differences in supply, I/O load, and measurement boundary are stated.

2 MCX E31 power management model

This section defines the application-visible Run and Standby power modes and distinguishes them from the internal power domains of the device. It summarizes mode behavior, retained resources, device differences, and the boundary for data sheet comparisons.

2.1 Run and Standby

[Table 1](#) compares the application-visible behavior of Run and Standby described in the MCXE31x Reference Manual ([MCXE31XRM](#)) and the SDK example [power_mode_switch_mcxe](#).

Table 1. Application-visible behavior in Run and Standby

Item	Run (FPM)	Standby (LPM)
Primary use	Executes application, communication, and computation	Wait for a defined wake-up event
Processor	Executes normally	Run domain is powered down and application execution stops
Clocks and peripherals	Enabled as required by the workload	Only Standby-domain and wake-up resources remain available
Software exit	Normal execution flow	Startup code runs again after wake-up

The `WFI` in `POWER_EnterStandbyMode()` is part of the final entry sequence, but it does not define an additional Device mode. Wake-up from Standby also differs from return from a shallow sleep. The Run domain powers up and the startup path executes again. The application must reconstruct the required state from Standby RAM, nonvolatile storage, or defaults. [Table 1](#) summarizes this transition model.

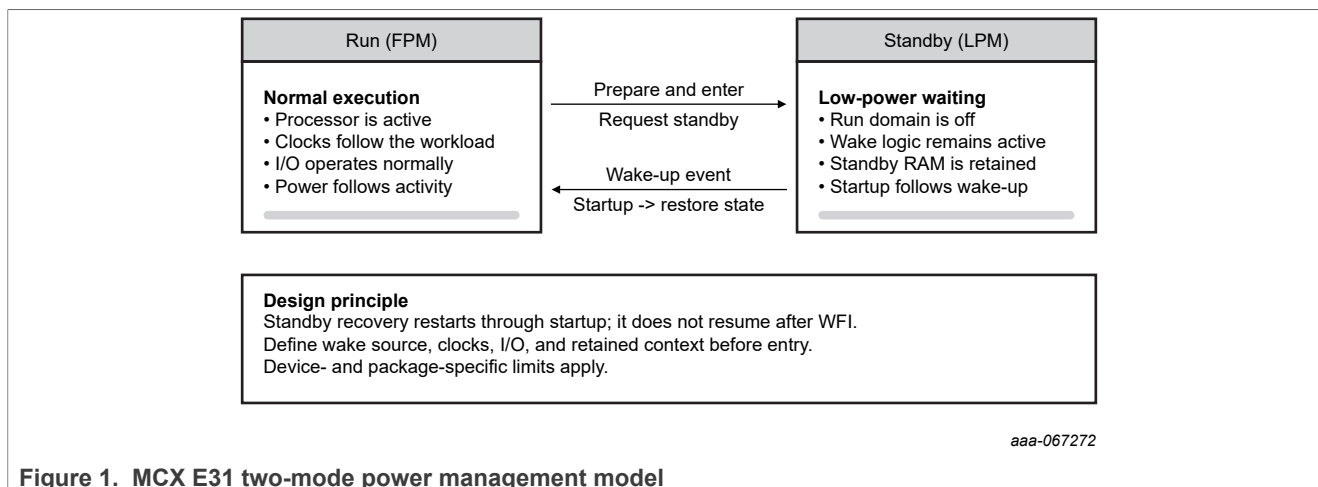


Figure 1. MCX E31 two-mode power management model

2.2 Retained resources and power domains

The Run domain contains the main execution resources and is powered down in Standby. The Standby domain retains the power-management and wake-up infrastructure needed to detect an event and start recovery. Important resources include the power-management controller, reset and wake-up logic, selected internal or external clock sources, RTC, SWT0, PIT0 RTI, LPCMP instances, and Standby RAM. For more details, see the *MCXE31x Reference Manual* ([MCXE31XRM](#)). [Table 2](#) links each resource to its design consequence.

Table 2. Standby-domain resources and design implications

Resource	Standby use	Design implication
Standby RAM	Retain selected application context	Retain only the state required to reconstruct the application
SIRC, FIRC, oscillators	Supply a selected Standby or wake-up function	Keep only the clock required by the wake-up contract
WKPU	Route internal and external wake-up events	Clear stale events before entry and after recovery
RTC, SWT0, PIT0 RTI	Generate timed wake-up events	Select according to time base, accuracy, and software ownership
LPCMP	Monitor an analog threshold or round-robin input	Include analog settling and input bias in the power budget
Pad keeping	Preserve or define selected I/O states	Configure every externally connected pin intentionally

2.3 Device differences

[Table 3](#) lists the data sheet device differences that affect this application note. For more details, see the *SIL2 robust 5V Arm Cortex M7 MCU Data Sheet* ([MCXEP172M160FB0](#)).

Table 3. MCX E31 device differences relevant to Standby

Item	MCX E315/316	MCX E317	MCX E31B
Maximum core frequency	120 MHz	120 MHz	160 MHz
Standby RAM	16 kB	32 kB	32 kB
FRDM procedure in this document	Not applicable	Not applicable	Applicable

The software concept applies across the family, but the board procedure must not be copied to a different device, package, or development board without checking its supplies, I/O domains, wake-up routing, and connections.

2.4 Boundary of data sheet comparisons

The MCX E31B data sheet lists the typical Standby supply currents in [Table 4](#). For more details, see the SIL2 robust 5V Arm Cortex M7 MCU Data Sheet ([MCXEP172M160FB0](#)). The data sheet marks these current data as preliminary and based on first samples.

Table 4. MCX E31B data sheet typical Standby supply currents at 25 °C

Standby condition	Typical VDD_HV_A current	Typical VDD_HV_B current
All clocks and peripherals off	50 µA	1.8 µA
SIRC on	52 µA	1.8 µA
FIRC on at 24 MHz	0.91 mA	1.8 µA

The data sheet conditions include $VDD_HV_A = VREFH = 5\text{ V}$, $VDD_HV_B = 5\text{ V}$, 25 °C, typical silicon, and no I/O load. The experiment in this document keeps SIRC enabled, retains the FRDM-MCX E31B board connections, and supplies VDD_HV_B at approximately 3.3 V. Therefore, the data sheet numbers are scale references, not compliance limits, or absolute characterization targets.

3 Standby software flow

This section maps the MCX E31 Standby requirements to the MCUXpresso SDK Low-level power mode switch example. It covers the entry sequence, startup-based recovery, clock and regulator handling, and Standby I/O configuration.

3.1 Entry sequence

The FRDM-MCX E31B low-level example performs the following sequence, consistent with the required reference-manual entry order. For more details, see MCX E31x Reference Manual ([MCXE31XRM](#)) and [power_mode_switch_mcxe](#):

1. Configure and clear the selected wake-up source.
2. Wait for debug-console transmission to complete, then deinitialize the console and UART pins.
3. Apply board-specific handling to the JTAG TDI pin.
4. Switch the system to 48 MHz FIRC as required for Standby entry, then disable PLL, FXOSC, SXOSC, and unused clocks.
5. Select which clock remains active in Standby.
6. Configure the power regulators and Standby I/O state.
7. Disable unused peripheral clocks and call `POWER_EnterStandbyMode()`.

[Figure 2](#) shows where the software preparation ends and startup-based recovery begins.

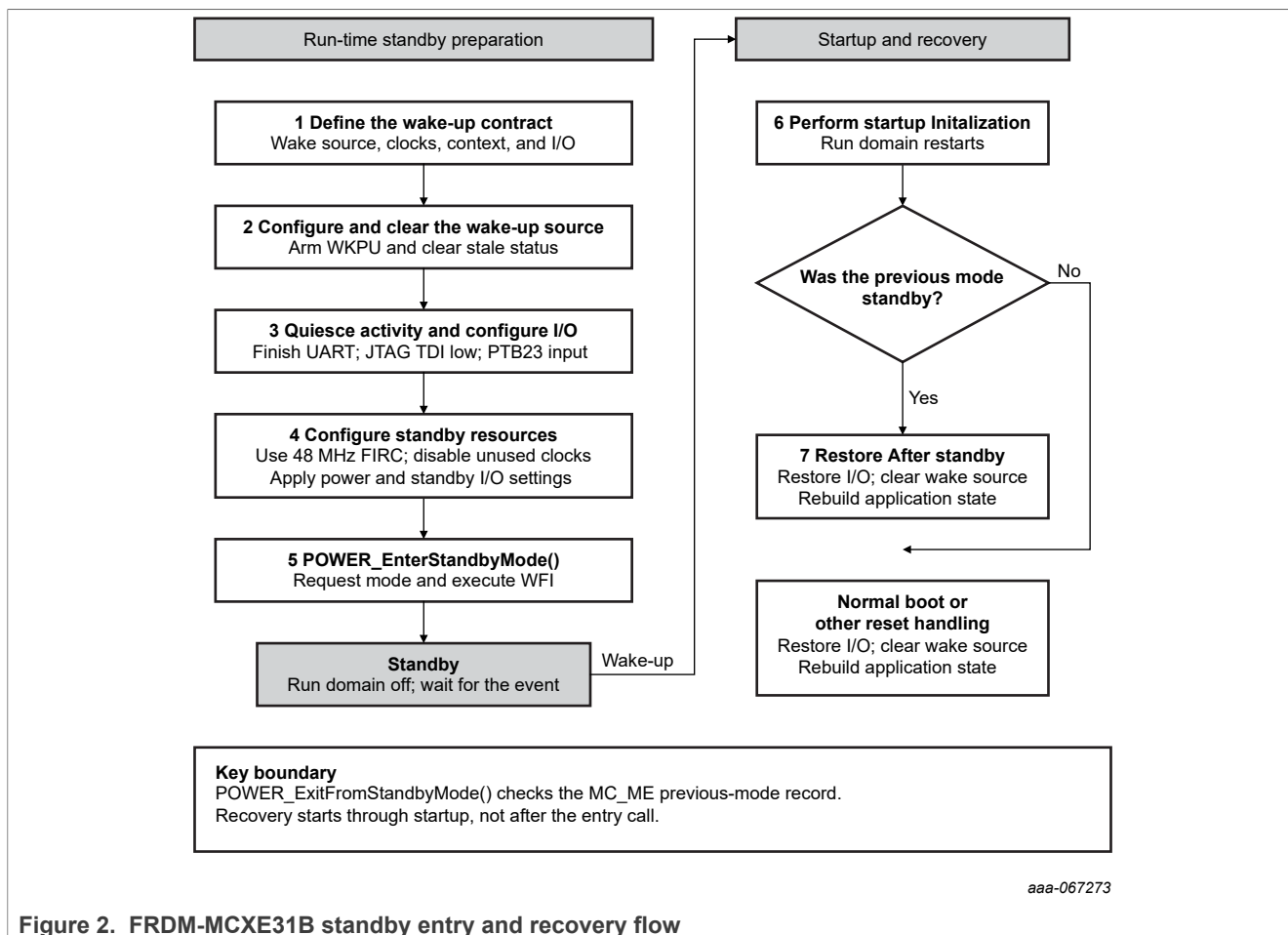


Figure 2. FRDM-MCX E31B standby entry and recovery flow

The 48 MHz FIRC used immediately before entering to Standby and the clock retained during Standby are separate decisions. The example uses FIRC while preparing the transition, then disables FIRC, and retains SIRC for the Standby interval.

3.2 Recovery path

After startup, the example calls `POWER_ExitFromStandbyMode()` to determine whether the previous mode was Standby, see the [power_mode_switch_mcx](#). The name can be misleading but the function identifies the recovery source. It is not the point at which hardware exits Standby. The example restores the I/O state, clears the wake-up request, deinitializes the wake-up peripheral, and prints:

```
Recover from standby mode.
```

The application then returns to the Run menu. A product must explicitly rebuild communication sessions, peripheral configuration, algorithm state, and scheduler state required after recovery.

3.3 Clock and regulator configuration

The unmodified SDK `APP_SetStandbyClockConfiguration()` uses:

```
CLOCK_DisableFircInStandbyMode();
CLOCK_EnableSircInStandbyMode();
```

It also disables unused high-frequency sources and the example disables the last-mile regulator, when present, and the low-power V25 regulator before the Standby request. Therefore, the measured configuration corresponds most closely to the SIRC-on row in the data sheet. It does not evaluate the all-clocks-off or FIRC-retained cases.

3.4 Standby I/O and pad keeping

I/O configuration is part of the power-state contract. Before the Standby request, software must decide whether each externally connected pin is actively driven, retained, used as a wake-up input, or placed in high-impedance.

The SDK example clears `DCM_GPR_DCMRWF1[STANDBY_IO_CONFIG]` before entry and sets it after recovery. This sequence enables pad keeping during Standby and removes it after recovery as required in the MCXE31x Reference Manual (MCXE31XRM) and [power_mode_switch_mcxe](#). The example also deinitializes the debug UART, pulls JTAG TDI to a defined state, and enables the input buffer for the selected WKPU button pin. Also, a product must review `SIUL2_MSCRn` settings for every connected pad:

- Use input-buffer and pull settings required by a wake-up input.
- Disable input and output buffers for an unused pin that must be high-impedance.
- Use pad keeping only when the retained static value is intentional.
- Avoid floating inputs and supply-domain mismatches that can dominate microamp-level measurements.

4 Wake-up and state-retention design

This section describes how to select a wake-up source and define the state that must survive Standby. It covers the SDK wake-up options, the wake-up contract, and the use of Standby RAM for retained context.

4.1 Wake-up sources in the SDK example

The low-level example exposes six wake-up choices. For more details, see the [power_mode_switch_mcxe](#). They demonstrate different paths through WKPU and the Standby domain; they are not interchangeable in a product. [Table 5](#) summarizes their intended use and main integration concern.

Table 5. Wake-up choices exposed by the SDK low-level example

Menu	Source	Suitable use	Main consideration
A	SWT0 timeout	Watchdog-supervised or demonstration timeout	Timeout ownership and watchdog servicing policy
B	RTC API timeout	Simple RTC-based delayed wake-up	API time base and RTC initialization
C	RTC timeout	Direct RTC wake-up path	Counter, alarm, and retained time state
D	LPCMP round-robin	Analog threshold monitoring	Input range, analog settling, and LPCMP current
E	PIT0 RTI	Periodic interval wake-up	RTI clock accuracy and interval
F	SW2 through WKPU	External event and open-ended current measurement	Pin bias, edge, filter, and contact behavior

The example routes SWT0, RTC API, RTC, LPCMP, and RTI events through WKPU sources 0 to 3, and SW2 through WKPU source 44 MCXE31x Reference Manual (MCXE31XRM) and [power_mode_switch_mcxe](#). The

measurement procedure uses SW2 so that the Standby plateau is not limited by a short software timeout. The functional test uses a five-second SWT timeout because it creates a deterministic recovery record.

4.2 Define the wake-up contract

Before implementing Standby, define:

1. The event that is allowed to wake the device.
2. The clock, analog block, pin state, or time base required by that event.
3. The context that must survive the transition.
4. The initialization sequence that reconstructs the application after startup.
5. The behavior when a wake-up event is already pending during entry.

Clear stale status before entry and record the recovery source early in startup. A product must not enable every possible wake-up source as a precaution; each retained resource increases integration complexity and can increase current.

4.3 Select what state to retain

[Table 6](#) provides a practical state-retention policy for startup-based recovery.

Table 6. Recommended state-retention policy

State type	Recommended location	Recovery action
Temporary calculation or communication state	Recreate after startup	Re-initialize the owning module or task
Small context required for fast resume	Standby RAM	Validate a version or signature, then restore
Configuration that must survive power loss	Nonvolatile memory	Load with an integrity and compatibility check
Default state	Constants or reset defaults	Reapply during normal initialization

MCX E315 and MCX E316 provide 16 kB of Standby RAM. However, MCX E317 and MCX E31B provide 32 kB. Linker placement is toolchain-specific. Usually, retaining an entire runtime image is less robust than retaining a small, versioned context, and rebuilding the rest.

5 Integration checklist and troubleshooting

This section provides a practical checklist for entering and leaving Standby and guidance for diagnosing unexpected current, failed entry, immediate wake-up, or incomplete recovery.

5.1 Before requesting Standby

Before requesting Standby, bring the software, wake-up path, I/O, retained context, and measurement setup to a known state. Complete the following checks before calling the Standby entry routine.

- Finish or cancel active transfers and wait for the debug UART transmission to complete.
- Configure and clear exactly one intended wake-up path.
- Move the system to the required entry clock before disabling PLL or oscillators.
- Disable unused peripheral clocks, analog blocks, and clock outputs.
- Define the state of every externally connected I/O and debug pin.
- Save only the context that is required after recovery.

- Verify that the current-measurement instrument cannot interrupt the supply during entry or wake-up.

5.2 After wake-up

Wake-up from Standby re-enters the startup path instead of returning from the entry call. Restore retained context, I/O, clocks, peripherals, and application state in a controlled order before resuming normal tasks.

- Detect the Standby recovery source before overwriting the retained context.
- Restore pad keeping and I/O configuration in the required order.
- to prevent an immediate repeated event, clear and deinitialize the wake-up source.
- Restore clocks before restoring peripherals that depend on them.
- Re-initialize communication, timing, and application state before resuming normal tasks.

5.3 Common reasons for unexpected current or failed entry

[Table 7](#) maps common symptoms to the first checks to perform.

Table 7. Standby integration troubleshooting guide

Symptom	Likely cause	Check
The current remains close to Run current	Standby request was rejected or code never reached <code>WFI</code>	Mode status, pending reset, console completion, and entry sequence
The current is tens of microamps higher than expected	Floating input, enabled clock output, active peripheral, or board pull device	SIUL2 settings, JTAG/SWD, LEDs, sensors, Ethernet, and expansion headers
The device resets when the meter changes range	Excessive burden voltage or an open supply path	Fixed range, temporary bypass, fuse, and lead connection
Immediate wake-up	Stale WKPU status or an active wake-up signal	Clear flags, input level, edge, and filter configuration
A recovery code runs but the application is invalid	The runtime state was assumed to survive	Standby RAM contents, context version, and reinitialization order
The measured current is negative or inconsistent	External I/O is back-powering a supply domain	I/O voltage levels and powered peripherals

6 Functional verification

The verified baseline follows the SDK example documentation [power_mode_switch_mcxe](#):

- FRDM-MCXE31B
- MCUXpresso SDK 26.06.00
- `boards/frdm-mcxe31b/demo_apps/power_mode_switch/low_level`
- 115200 bit/s, 8 data bits, no parity, 1 stop bit, and no flow control

The unmodified example is programmed to the board. An SWT timeout is selected and the device recovered after five seconds. The relevant terminal transcript is:

```
Standby: Low Performance mode...
SWT is selected as wakeup source!
Will wakeup in 5 seconds.
Recover from standby mode.
Power mode: Run in FPM
```

This record confirms the functional Run-to-Standby-to-startup path. It does not by itself prove a particular Standby current. For the steady-current measurement, option F, SW2/WKPU, is used to keep the Standby plateau active for the measurement window. No test required any changes to the source code.

7 FRDM-MCXE31B current experiment

This section documents a board-level current experiment using the unmodified FRDM-MCXE31B SDK example. It defines the measurement boundary, equipment, sequential JP1 and JP2 procedure, results, and interpretation limits.

7.1 Objective and measurement boundary

The experiment records one stable average current at JP1 and one at JP2 using the unmodified SDK binary. No board resistor or supply voltage is changed.

JP1 is in the P5V0 to VDD_HV_A_MCU path. Its current includes the VREFH_MCU path supplied through R6. JP2 is in the P3V3_BOARD to VDD_HV_B_MCU path. For more details, see the FRDM-MCXE31B Schematics, SPF-95490. The two rail currents are reported separately. The values are not the total USB input current. Also, a single value does not represent the complete MCU current.

7.2 Equipment and conditions

Table 8 records the conditions that bound the reported measurements.

Table 8. Equipment and recorded measurement conditions

Item	Recorded condition
Board and software	FRDM-MCXE31B, MCUXpresso SDK 26.06.00 unmodified low-level example
Wake-up source	SW2 through WKPU
Instrument	Fluke 17B+ digital multimeter, DC μ A range
Stable window	Approximately 20 seconds after approximately 5 seconds of settling
Ambient	Room temperature, not controlled
Board connections	MCU-Link USB connected; no expansion board attached

A temporary low-resistance bypass across the shunt under test carries Run startup and wake-up current. It is removed only after the application enters Standby and reconnected before SW2 is pressed. This prevents a normal microamp range from interrupting the MCU supply during a high-current transition.

7.3 Connection and measurement procedure

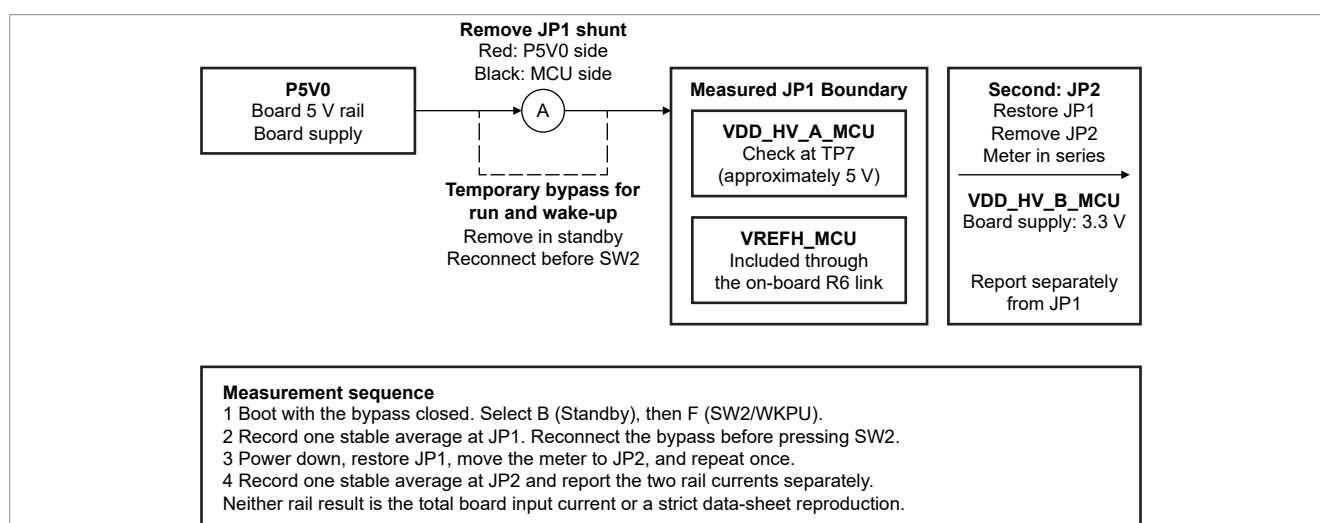
Measure JP1 and JP2 sequentially because each shunt represents a different MCU supply rail. Keep a temporary low-resistance bypass in place during startup and wake-up, and remove it only after Standby is stable.

1. Power off the board, remove JP1, and leave JP2 fitted. Connect the meter from the P5V0 side of JP1 to the VDD_HV_A_MCU side.
2. Close the temporary bypass. Power the board. Select B for the Standby, then F for SW2/WKPU wake-up.
3. After the final terminal message, select the DC μ A range. Remove the bypass and allow the reading to settle. Record an average over approximately 20 seconds.

4. Reconnect the bypass before pressing SW2. Confirm that the Run menu returns and then power off the board.
5. Restore JP1, remove JP2, and connect the meter from the P3V3_BOARD side of JP2 to the VDD_HV_B_MCU side. Keep the onboard 3.3 V supply; do not apply external 5 V.
6. Repeat the same Standby and measurement sequence for JP2, then restore both shunts after power is removed.

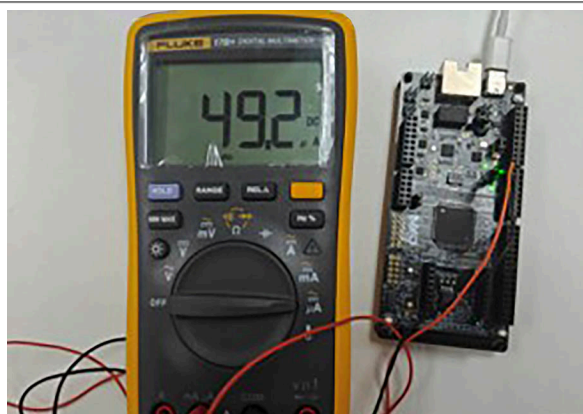
Never connect a meter in current mode directly between a supply and the ground. Power down before moving a shunt or meter lead, and do not remove JP1 and JP2 at the same time.

Figure 3 shows the sequential connections and the temporary bypass used during Run and wake-up transitions.

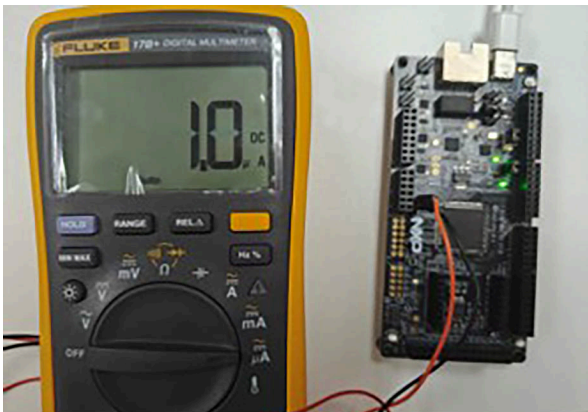


aaa-067274

Figure 3. Sequential JP1 and JP2 current measurements on FRDM-MCX E31B



aaa-067347



aaa-067348

Figure 4. Representative instantaneous readings during the JP1 and JP2 Standby measurements

Figure 4 shows 49.2 μA at JP1 and approximately 1.0 μA at JP2 at the instant each photograph is taken. The result table reports the average over the stable window; a photographed instantaneous reading can differ by one least significant digit.

7.4 Results and interpretation

Table 9 reports the two rail measurements and preserves the applicable data sheet boundary for each comparison.

Table 9. FRDM-MCX E31B Standby rail-current results

Rail	Measurement boundary	Measured supply	20 s average	Data sheet context	Interpretation
JP1	VDD_HV_A_MCU + VREFH_MCU	4.97 V	49.2 μA	52 μA typical, SIRC on	Same order under the recorded board conditions; not a characterization comparison
JP2	VDD_HV_B_MCU	3.27 V	1.1 μA	1.8 μA typical at 5 V	Plausible board result; not a direct error comparison because the supply voltage differs

The JP1 result is close to the data sheet SIRC on typical value, but it remains a board-level observation rather than a characterization result. The JP2 result is lower than the 5 V data sheet typical value while the board supplies the domain at 3.27 V. Supply voltage, silicon distribution, uncontrolled room temperature, instrument behavior, I/O state, and board connections all limit a direct comparison.

The two currents are measured sequentially at different supply boundaries. They must not be added and presented as a complete MCU or board current. The experiment demonstrates a practical measurement method and confirms that the unmodified SDK reaches a low-current Standby state on the tested board.

8 System-level power interpretation

The lowest Standby current does not necessarily produce the lowest average energy. For a periodic task, a first-order estimate is:

$$I_{_AVG} = (I_{_RUN} \times t_{_RUN} + I_{_STANDBY} \times t_{_STANDBY}) / (t_{_RUN} + t_{_STANDBY})$$

When supply voltages differ, calculate the power for each rail: $P_{_AVG} = \text{sum}(V_{_i} \times I_{_AVG_i})$. Include entry and recovery energy when the Standby interval is short. A final product decision must compare energy per useful operation and event period, not only one mode current.

9 Conclusion

MCX E31x low-power software starts with a defined wake-up contract and an ordered transition of peripherals, I/O, clocks, regulators, and retained state. Standby wake-up follows the startup path, so robust applications retain a small explicit context and rebuild the rest.

The MCUXpresso SDK low-level example provides a directly runnable reference flow and retains SIRC in its default Standby configuration. On the tested FRDM-MCXE31B, the unmodified example recovered correctly through both timed and button wake-up paths. Board-level measurements produced 49.2 μA at the JP1 $VDD_HV_A + VREFH$ boundary and 1.1 μA at the JP2 VDD_HV_B boundary.

These results demonstrate the method and expected scale; they do not reproduce all data sheet characterization conditions. A product must repeat the review for its own supplies, I/O loads, wake-up sources, retained context, temperature range, and energy profile.

10 Acronyms

[Table 10](#) lists the acronyms used in this document.

Table 10. Acronyms

Acronym	Definition
DCM	Device Configuration Module
DS	Data Sheet
FIRC	Fast internal reference clock
FPM	Full-Power Mode; the SDK menu describes it as Full Performance mode
IO	Input/Output
JTAG	Join Test Action Group
LPCMP	Low-Power Comparator
LPM	Low-Power Mode
PMC	Power-Management Controller
RTC	Real-Time Clock
RTI	Real-Time Interrupt Timer
SDK	Software development kit
SIRC	Slow Internal Reference Clock
SWT	Software Watchdog Timer

Table 10. Acronyms...continued

Acronym	Definition
WKPU	Wake-Up Unit

11 References

- SIL2 robust 5V Arm Cortex M7 MCU Data Sheet ([MCXEP172M160FB0](#))
- MCXE31x Reference Manual ([MCXE31XRM](#))
- [power_mode_switch_mcxe](#)
- FRDM-MCXE31B Schematics, SPF-95490
- Low power implementation on MCXA156 MCU series ([AN14447](#))
- Power Management for MCX E24x ([AN14736](#))

12 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2026 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

13 Revision history

[Table 11](#) summarizes the revisions to this document.

Table 11. Revision history

Document ID	Release date	Description
AN15135 v.1.0	21 September 2026	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Contents

1	Introduction	2
2	MCX E31 power management model	2
2.1	Run and Standby	2
2.2	Retained resources and power domains	3
2.3	Device differences	3
2.4	Boundary of data sheet comparisons	4
3	Standby software flow	4
3.1	Entry sequence	4
3.2	Recovery path	5
3.3	Clock and regulator configuration	5
3.4	Standby I/O and pad keeping	6
4	Wake-up and state-retention design	6
4.1	Wake-up sources in the SDK example	6
4.2	Define the wake-up contract	7
4.3	Select what state to retain	7
5	Integration checklist and troubleshooting	7
5.1	Before requesting Standby	7
5.2	After wake-up	8
5.3	Common reasons for unexpected current or failed entry	8
6	Functional verification	8
7	FRDM-MCXE31B current experiment	9
7.1	Objective and measurement boundary	9
7.2	Equipment and conditions	9
7.3	Connection and measurement procedure	9
7.4	Results and interpretation	11
8	System-level power interpretation	12
9	Conclusion	12
10	Acronyms	12
11	References	13
12	Note about the source code in the document	13
13	Revision history	13
	Legal information	14

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.